

Szegedi Tudományegyetem
Informatikai Tanszékcsoporth

**Java metódusok karbantarthatóságát jelző
modellek építése szakértői vélemények alapján**

Országos Tudományos Diákköri Dolgozat

Készítette:

Ladányi Gergely

II. évf. Programtervező Informatikus MSc

Témavezetők:

Dr. Ferenc Rudolf

Egyetemi adjunktus

Hegedűs Péter

Doktorandusz

Szeged

2012

Tartalomjegyzék

1. Bevezetés	4
2. Kapcsolódó munkák	5
2.1. ISO/IEC 9126-os szabvány	5
2.2. Metrika alapú szoftverminőség mérés	6
3. Felmérések, eredmények előállítása	9
3.1. Metrika Kiértékelő Keretrendszer	9
3.2. Különböző felmérések	11
3.2.1. Szakértői kiértékelés	11
3.2.2. Egyszemélyes kiértékelés	11
3.2.3. Hallgatói kiértékelés	11
4. Eredmények	13
4.1. A dolgozat eredményeit összefoglaló kérdések, és azok jelentősége	13
4.2. Alkalmazott forráskód metrikák és a különböző felmérések véleményei- nek kapcsolata	14
4.2.1. Az alkalmazott metrikák	14
4.2.2. Kapcsolat erősségének összehasonlítása regressziós eljárások sze- rint	15
4.2.3. Kapcsolat erősségének összehasonlítása felmérések szerint	17
4.3. Gépi tanulási módszerek hatékonysága a különböző felmérések esetén . .	19
4.3.1. Alkalmazott gépi tanulási módszerek	19
4.3.2. Különböző algoritmusok hatékonyságának összehasonlítása	20
4.3.3. Felmérések szerinti összehasonlítás	21
4.3.4. Többváltozós lineáris regresszió a metrikák logaritmusán	23
4.3.5. Modellek elemzése	24
4.3.6. Eredmények összefoglalása	26
4.4. Hallgatók teljesítménye és a szakértői véleményekhez viszonyított pon- tatlanságának kapcsolata	27
5. Összefoglalás, jövőbeli elképzelések	28
6. Köszönetnyilvánítás	30
Köszönetnyilvánítás	30

7. Mellékletek	33
7.1. Alkalmazott metrikák leírása	33
7.2. Vélemények és metrikák közötti korrelációs értékek	34

1. Bevezetés

Napjainkban a szoftverfejlesztés egyik központi témája a szoftverrendszerek karbantartóságának vizsgálata, amely leginkább a szoftverek fejlesztési költségére és megbízhatóságára gyakorolt hatásának köszönhető. A nehezen karbantartható rendszerek fejlesztési költségei sokkal magasabbak, valamint sokkal valószínűbb, hogy nem várt hiba adódik a rendszer működése során, amely kritikus jelentőségű lehet az élet számos területén, például légi irányításnál, banki rendszereknél, stb. Ezen okokból kifolyólag nagyon fontos azonosítani a rendszerek rosszul karbantartható metódusait. A karbantarthatóság egy-egy tanulmányozhatóságának érdekében megalkotott ISO/IEC 9126 [11] nemzetközi szabványban definiálták, hogy a karbantarthatóságot milyen alacsonyabb szintű tulajdonságok befolyásolják. Ezen tulajdonságok azonban szubjektívek, ilyen pl. a változtathatóság, tesztelhetőség, stb. A metódusok ezen tulajdonságainak meghatározáshoz így emberi kiértékeléseket használtam, amelyek alapján azután a forráskód metrikákat prediktorként felhasználva az emberi véleményt közelítő modelleket tudtam építeni gépi tanulási módszerek segítségével.

A dolgozat célja tehát a különböző metódusok karbantarthatóságának meghatározása a szabvány iránymutatásai alapján emberi kiértékelések segítségével, illetve a különböző kiértékelések alapján épített tanuló modellek hatékonyságának összehasonlítása, valamint a különböző metrikák és vélemények közötti kapcsolatok feltárása. A felmérések során hozzávetőlegesen 300 ember 150000 kérdésre adott válasza alapján épített különböző gépi tanulási modellek a karbantarthatóságot például 0.72-es Pearson korrelációval, és 0.83-as átlagos abszolút hibával közelítették [0,10]-es folytonos skálán, ahol a 0 jelenti a teljesen karbantarthatatlan metódust, míg a 10 pedig azt, hogy a metódus tökéletesen karbantartható.

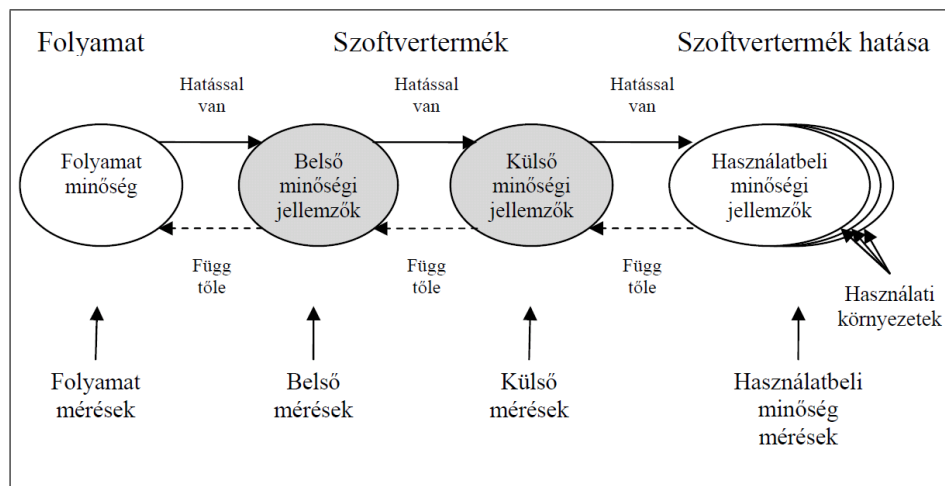
A dolgozat célja az alábbi kérdések kísérleti eredményekkel történő megválaszolása:

1. Milyen a kapcsolat jellege és erőssége az egyes felmérések szubjektív véleményei és a különböző forráskódmetrikák között?
2. Milyen hatékonyan képesek a jól ismert forráskód metrikák alapján az egyes gépi tanulási módszerek megjósolni a metódusokra érkező véleményeket?
3. Az alkalmazott metódus-véleményezési technikák milyen hatékonyságbeli eltérést mutatnak?
4. Hogyan függ össze a hallgatók egyetemi teljesítménye és a metódusokra adott válaszaik szakértőkhöz viszonyított pontossága?

2. Kapcsolódó munkák

2.1. ISO/IEC 9126-os szabvány

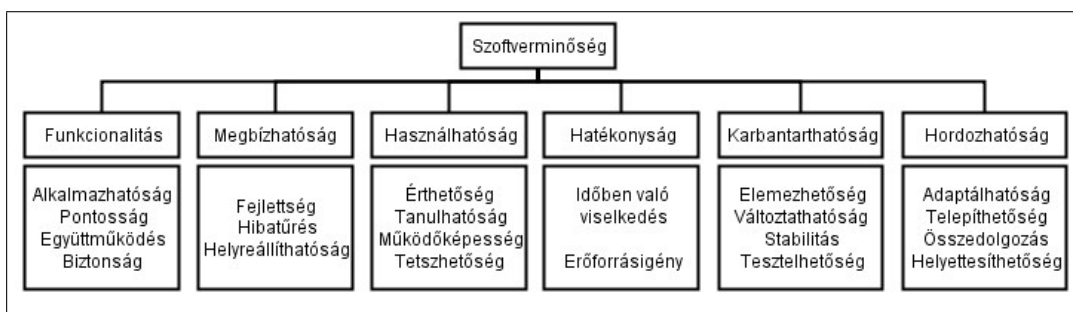
Egy olyan általános jellemző meghatározása során, mint a szoftverminőség elengedhetetlen az egységes szemléletmód kialakítása, hiszen így a cél elérése érdekében tett erőfeszítéseket közös nevezőre lehet hozni. Éppen ezért azokat a szempontokat, melyeket a szoftver minőségének meghatározása során érdemes figyelembe venni különböző minőségi modellekkel írhatjuk le, mint például: ISO/IEC 9126 [11], ISO/IEC 25000/SQuaRE [12], ISO/IEC 15504/SPICE [10], Automotive SPICE [1] és CMMI [5]. Ezen modellek közül a legelterjedtebb az ISO/IEC 9126, így a felmérésekben használt kérdéseket ez alapján állítottuk össze.



1. ábra. Az ISO/IEC 9126-os szabvány felépítése [18]

A modell az 1. ábrán is látható hierarchikus felépítéssel rendelkezik, amely szerint a fejlesztés során alkalmazott folyamat hatással van a rendszer belső minőségi jellemzőire. Ezek a belső minőségi jellemzők közvetlenül a rendszer forráskódszintű minőségét jellemzik és hatással vannak a külső minőségi jellemzőkre, melyeknek jelentése annyiban speciálisabb, hogy már függ a rendszer aktuális környezetétől, pl. futás közbeni hibák mérése. Végül ezen külső minőségi jellemzők hatással vannak az úgynevezett használatbeli minőségi jellemzőkre, melyek már felhasználói szemszögből jellemzik a rendszert.

A belső és külső jellemzőkhöz egyaránt a 2. ábrán látható tulajdonságok tartoznak (funktionalitás, megbízhatóság, használhatóság, hatékonyság, karbantarthatóság, hordozhatóság), valamint minden tulajdonsághoz felsoroltam a hozzá tartozó altulajdonságokat is. A modell tulajdonságai közül a költségekre gyakorolt hatásának köszönhetően legjelentősebbnek tartott a karbantarthatóság, így ennek meghatározásával foglalkozunk.



2. ábra. Az ISO/IEC 9126-os szabványban definiált tulajdonságok, altulajdonságok

A modell a karbantarthatóság következő altulajdonságait nevezi meg:

1. *Elemezhetőség*: a szoftver azon képessége, hogy a szoftver hiányosságai és különböző hibák okai milyen könnyen azonosíthatóak.
2. *Változtathatóság*: a szoftver azon képessége, amely lehetővé teszi a meghatározott változtatások végrehajtását.
3. *Stabilitás*: a szoftver azon képessége, hogy elkerülje a módosítások következtében fellépő, előre nem látható hatásokat.
4. *Tesztelhetőség*: a szoftver azon képessége, hogy lehetővé tegye a rajta végrehajtott módosítások ellenőrzését.
5. *Karbantarthatósági megfelelés*: a szoftver azon képessége, hogy betartja a karbantarthatósághoz kapcsolódó szabványokat, konvenciókat.

Ezek mellett fontosnak tartottuk felvenni az altulajdonságok közé az érthetőséget, hiszen egy rendszer karbantartása során jelentős időt lehet megtakarítani, ha gyorsan fel lehet térképezni a rendszert, gyorsan meg lehet érteni a kód működését.

6. *Érthetőség*: mennyire könnyű a kód algoritmusának működését megérteni, átlátni (azaz mekkora a megértéshez szükséges időtartam hossza).

2.2. Metrika alapú szoftverminőség mérés

A ISO/IEC 9126-os szabvány definiálja, hogy milyen tulajdonságokat értsünk a szoftver minősége alatt, arra azonban már nem tér ki miképpen érdemes azokat meghatározni, ezért a gyakorlatban számos egymástól különböző megoldás született. A legtöbbször, hozzánk hasonlóan statikus forráskódelemzés eredményei alapján próbálták megoldani a problémát.

A statikus forráskódelemzésen alapuló minőségbiztosítás célja a karbantarthatósági költségek csökkentése, valamint a szükséges javítások jelzése. Jelentősége abban rejlik, hogy folyamatos alkalmazása mellett még a fejlesztési fázisban számos olyan problémára derülhet fény, melyek javítási költsége az idő előrehaladtával egyre nő, és egyre nagyobb valószínűséggel okoz hibás működést. A rendszer elemzése során a forráskód alapegységeihez (osztályaihoz, metódusaihoz) olyan mérőszámokat, más néven metrikákat rendelünk, melyek számos tekintetben (pl. méret, komplexitás) alacsony szinten jellemzik őket. Azonban az ilyen alacsony szintű metrikák önmagukban nem képesek meghatározni szubjektív, magas szintű tulajdonságokat, ezáltal szükségesség válik azok, valamilyen módszerrel történő kombinálása, aggregálása. A gyakorlatban sokszor előfordul például, hogy egy rendszert egyetlen magas szintű mérőszámmal kell jellemezni, amely alapján a rendszer minősége elhelyezhető más rendszerekhez képest. A következőkben áttekintem az irodalomban talált, a dolgozathoz kapcsolódó eredményeket, kihangsúlyozva a hasonlóságokat, különbségeket.

Az irodalom ismertetését a dolgozat előzményének számító munkával [16] kezdem, melyben Hegedűs Péter témavezetőm vezetésével metrikák segítségével közelítettük a karbantarthatósági altulajdonságokat. Meghatároztuk a metrikák és az egyes altulajdonságok közötti lineáris kapcsolat erősségét, valamint a szakértői vélemények alapján osztályozás segítségével becsültük az egyes altulajdonságokat. A metódusok eltérő szakértői megítéléséből adódó szórásból kifolyólag nem sikerült elég hatékony modellt alkotni. A tapasztalatok alapján a felméréseket, tanulási folyamatokat újragondolva hatékonyabb eljárásokat dolgoztam ki. A szórás csökkentésében jelentős javulást értünk el az egyszemélyes felmérés alkalmazásával, majd később az egy metódusra érkező hallgatói válaszok átlagának globális véleményként történő kezelésével. A jelen dolgozatban a szakértői, hallgatói és egyszemélyes felmérések alapján regressziós tanulás segítségével állítottam elő a becsléseket, ezáltal az egyes felmérések hatékonyságát össze is tudtam hasonlítani. A metrikák és az altulajdonságok közötti kapcsolat vizsgálata során a valóságot jobban leíró exponenciális és logaritmikus regressziós eljárásokat is alkalmaztam.

Stefan Wagner és mások [20] egy széleskörű felmérés eredménye alapján megállapították, hogy a cég-specifikus minőségi modelleket gyakran különböző szabványok (például az ISO/IEC 9126) segítségével határozzák meg. A felmérés rámutatott, hogy milyen fontosak a különböző minőségi modellek a tesztelés, kódelemzés és minőségmérés során. Ezek alapján kijelenthető, hogy a modell élesben történő használatakor, az előállt eredmények értelmezése során elengedhetetlen az egyes tulajdonságok és metrikák közötti kapcsolat erősségének és jellegének ismerete.

Jung és Kim [14] az ISO/IEC 9126 szabvány egyes altulajdonságai közötti kapcsola-

tot tanulmányozták egy felmérés során. Az altulajdonságokat csoportokba rendezték egymással való kapcsolatuk erőssége alapján, melyet a felmérésekre érkező válaszok alapján számoltak ki. Arra a következtetésre jutottak, hogy az így kialakult csoportok nagy hasonlóságot mutattak a szabványban definiált tulajdonságokkal. A modell jelentőségét ez a felmérés is alátámasztotta, ezért mi már nem az altulajdonságok és a tulajdonságok között kerestünk kapcsolatot, hanem az altulajdonságok és a metrikák között.

A különböző forráskód metrikák közötti kapcsolat feltárásával számos tanulmány foglalkozott már, azonban legtöbbször ezt osztály szinten végezték [4] [7]. Velük ellentétben mi metódus szinten vizsgáltuk a szoftver minőségét, hiszen így pontosabban tudunk rámutatni a rendszerben található problémákra. A dolgozatban csak említésképpen tüntettem fel a metrikák közötti korrelációs értékeket, a hangsúlyt a metrikák és a szubjektív tulajdonságok közötti kapcsolatok feltárására helyeztem.

A metrikákat a rendszer számos más tulajdonságának meghatározására is használták már, például Hector M. Olague és mások [15], valamint Gyimóthy, Ferenc és Siket [7] regressziós és gépi tanulási módszerek segítségével tettek kísérletet a hibás forráskódelemek meghatározására. A dolgozatban hozzájuk hasonlóan én is regressziós és gépi tanulási módszereket használtam, azonban én nem a hibás elemek megtalálására, hanem a metódusok szabványban definiált altulajdonságainak meghatározására használtam őket.

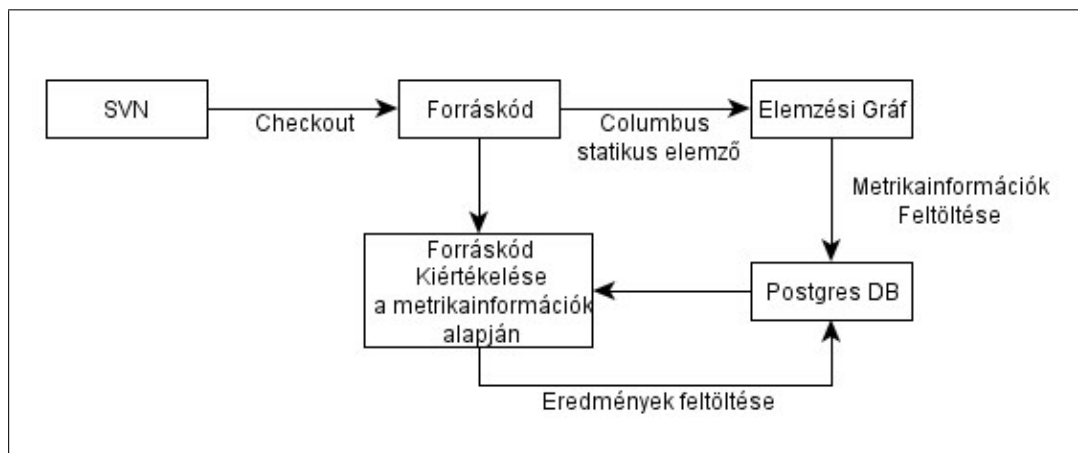
Számos módszertan, keretrendszer született már az ISO/IEC 9126-os szabványon alapuló rendszerszintű karbantarthatóság megállapítására. Ezen módszertanok is metrikák és szakértői vélemények segítségével számolják ki a karbantarthatóságot, a különbség ezen adatok aggregálásában, felhasználásában mutatkozik. T. Bakota és mások [2] egy valószínűségszámításon alapuló módszert mutatnak be a rendszer karbantarthatóságának meghatározására. Egy benchmark rendszerekből álló nagyméretű adatbázis és a metrikák eloszlásának segítségével hatékonyan közelítették a szubjektív szakértői véleményeket. Heitlager, Kuipers és Visser [9] szintén egy karbantarthatósági modellt mutattak be. A metrikaértékeket különböző kategóriákba sorolták, majd ezen kategóriák segítségével aggregálták a karbantarthatóságot. Bansiya és Davis [3] egy hierarchikus modell (QMO-OD) segítségével számolták ki a karbantarthatóságot. Ezen modellekhez hasonlóan én is az ISO/IEC 9126 szabvány szerint vizsgáltam a karbantarthatóságot, azonban a céloom nem rendszerszinten, hanem a metódusszinten való közelítése volt.

3. Felmérések, eredmények előállítása

Ebben a fejezetben bemutatom a felmérésekhez használt rendszert, majd röviden ismertetem az egyes felmérések jellegzetességeit.

3.1. Metrika Kiértékelő Keretrendszer

A metódusok kiértékelését a Szegedi Tudományegyetem Szoftverfejlesztés Tanszéken fejlesztett Metrika Kiértékelő Keretrendszer segítségével végeztem. A keretrendszer működését ábrázoltam a 3. ábrán. A beüzemelés első lépése tehát a kiértékelni kívánt rend-



3. ábra. Metrika Kiértékelő Keretrendszer felépítése

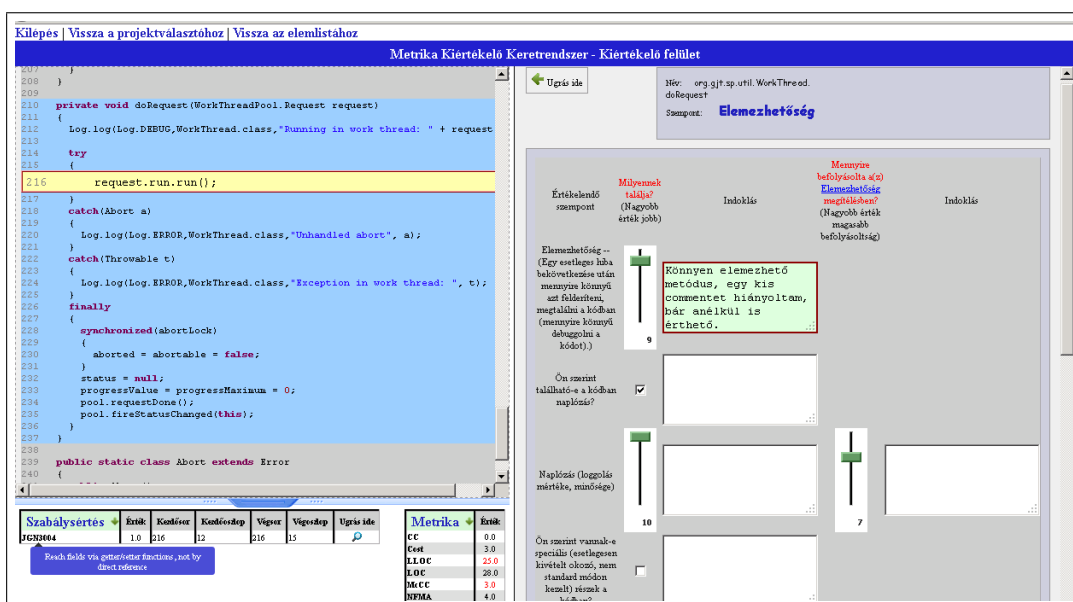
szet elérési útjának (pl. SVN repository) megadása. Majd a rendszer letöltése után a Columbus [6] statikus elemző segítségével minden metódushoz kiszámolásra kerülnek annak forráskód metrikái, előáll egy a rendszer összes metódusának metrikaértékeit tartalmazó hierarchikus elemzési gráf. Az elemzési gráfban található információk adatbázisba való feltöltése után a kiértékelő elkezdheti a metódusok forráskódjának kiértékelését. A felhasználó kiválasztja a számára elérhető metódusok valamely tulajdonságát és a 4. ábrán látható módon kiértékeli azt. A rendszer a metódus alatt kilistázza a hozzá tartozó metrikaértékeket és megmutatja a kódban fellelt szabálysértéseket is. A kiértékelés első kérdése mindig az adott tulajdonság milyenségére kérdez rá, ezeket az értékeket fogjuk közelíteni gépi tanulási módszerekkel az eredmények fejezetben.

Egy adott tulajdonság esetén számos egyéb kérdést is feltettünk a kiértékelőnek. Fel sorolás szinten minden tulajdonsághoz megadok egy-egy ilyen tulajdonságspecifikus kérdést.

- Elemezhetőség: *Milyennek találja a Naplózást (loggolás mértéke, minősége)?*

- Változtathatóság: *Milyennek találja a metódusok számosságát, elnevezését (például egy `f()` hívás sokkal kevésbé érthető elsőre, mint egy `getName()` hívás)?*
 - Stabilitás: *Ön szerint találhatók-e a kódban olyan veszélyes szerkezetek, amikre nem jött szabálysértés?*
- Ez egy eldöntendő kérdés, amennyiben nemleges választ kaptunk, azonnal eltűnik a következő kérdés, ami veszélyes szerkezetek milyenségére kérdez rá, hiszen nincs is olyan kódban, nincs miről kérdezni.
- Tesztelhetőség: *Milyennek találja a kód környezetének szimulálhatóságát?*
 - Érthetőség: *A kód méretéhez, funkcionalitásához, bonyolultságához képest milyennek találja a kommentezés arányát?*

Minden tulajdonság esetén a fentiek mellett megkérdeztük, hogy lehet-e javítani a metódus adott tulajdonságán és mekkora erőforrásráfordítás lenne szükséges hozzá. Az egyes tulajdonságokat mindig a 0,1,2..,10-es skálán értékelhették a felhasználók, amely skálán 0 jelenti például a teljesen értelmezhetetlen metódust, míg a 10-es a tökéletesen értelmezhető.



4. ábra. Metrika Kiértékelő Keretrendszer használat közben

A rendszer folyamatos evolúción esett át a kiértékelők visszajelzései alapján. Kezdetben túl sok kérdést tettünk fel egyenként a kiértékelőknek és minden válaszukat indokolni is kellett. Ezáltal túlságosan hosszadalmassá és így költségessé vált egy metódus kiértékelése. Csökkenteni kellett a kérdések számát, valamint gyorsítani a kiértékelést azáltal,

hogy a kérdésekre a kiértékelők egy oldalon válaszolhatnak és csak akkor kellett indokolniuk, ha valamilyen problémát észleltek a metódussal kapcsolatban. Az én feladatom az előálló adatok elemzése mellett ennek a gyorsításnak az implementálása volt.

3.2. Különböző felmérések

2011-ben három különböző, egymástól lényegében eltérő felmérést végeztünk metódusok kiértékelésére. A felmérések során hozzávetőlegesen 150000 kérdésre kaptunk választ ($300 \text{ ember} * 10 \text{ metódus} * 5 \text{ tulajdonság} * 10 \text{ kérdés a metódus adott tulajdonságáról}$). Amennyiben valamely tulajdonságra rossz értékelést adott a kiértékelő, akkor indokolnia kellett. Minden metódus során megkérdeztük annak 5 tulajdonságát (elemezhetőségét, változtathatóságát, tesztelhetőségét, stabilitását, érthetőségét). A hallgatói kiértékelés során felkerült a kérdések közé a karbantarthatóság is önmagában.

3.2.1. Szakértői kiértékelés

Az első kiértékelést a Szoftverfejlesztés tanszéken dolgozó szoftverminősítéssel foglalkozó 34 tapasztalt programozó végezte. A jEdit [13] nyílt forráskódú rendszer 350 metódusának számos tulajdonságát értékelték ki úgy, hogy minden kérdésre egy vélemény érkezett. A felmérés eredménye arra mutatott rá, hogy a tulajdonságok megítélésében nagy szórás mutatkozik, éppen ezért a gépi tanulás során nem tudott elég hatékony modell felépülni.

3.2.2. Egyszemélyes kiértékelés

A vélemények nagymértékű szórásának kiküszöbölésére tett első kísérletünk során egy tapasztalt szakértő kiértékelésére próbáltam meg modellt építeni. A felmérés során egy zárt forráskódú rendszer 250 metódusát értékelt ki egy 2 éves tapasztalattal rendelkező programozó. A kiértékelés alapján az elvártaknak megfelelően hatékonyabb modell állt elő, azonban a modell által elég hatékonyan megjósolt vélemény nem tekinthető reprezentatív véleménynek.

3.2.3. Hallgatói kiértékelés

A hallgatói kiértékelésben hozzávetőlegesen 200 haladó Java előismeretekkel rendelkező hallgató vett részt. A nagy létszámú felmérés lehetővé tette 200 metódus olyan jellegű kiértékelését, mely során minden metódusra legalább 7 vélemény áll rendelkezésre. Azokat a metódusokat, melyekre kevesebb vélemény érkezett elvetettük (metódusok 10%-a).

Minden módszerre vettük a vélemények átlagát, ezzel közelítve a módszer egyes tulajdonságainak globális hallgatói megítélését. Átlag helyett használhattunk volna mediánt is, azonban egy kicsivel kevésbé hatékony modelleket kaptunk ezzel a módszerrel.

A felmérés jellegéből következően számszerűsíteni is tudtam, hogy egy módszer egy adott tulajdonságának megítélésében mekkora szórás mutatkozik. A kapott eredményeket összefoglaló 1. táblázatból leolvasható, hogy az egyes tulajdonságok módszershintű megítélésében körülbelül 2 egységnyi szórás mutatkozik.

1. táblázat. Az egyes tulajdonságok megítélésének átlagos szórása

Tulajdonság	Szórás
Elemezhetőség	1.859
Változtathatóság	2.049
Stabilitás	2.222
Tesztelhetőség	2.019
Érthetőség	1.880
Karbantarthatóság	1.975

Egy [0,10]-es skálán ez meglepően nagy érték, megmagyarázza miért nem lehetett elég hatékony modellt építeni a szakértői kiértékelés eredményei alapján. Ugyanakkor érdemes megjegyezni, hogy valószínűleg a szakértői vélemények esetén kisebb lehet a szórás, hiszen ha feltesszük, hogy létezik egy objektív karbantarthatósági értéke egy módszernek, akkor a szakértői vélemények várhatóan kisebb szórással mozognak az érték körül, mint a hallgatói vélemények. Sajnos ezzel kapcsolatos mérőszám előállításához nagyszámú szakértői kiértékelésre lenne szükség, ami igen költséges lenne.

4. Eredmények

A dolgozat fő célja tehát a különböző metódusokra adott vélemények és metrikaértékek alapján felépített gépi tanulási modell segítségével meghatározni az ISO/IEC 9126 modellben definiált karbantarthatósági altulajdonságokat. Ezen cél elérése érdekében tett erőfeszítések közben több érdekes eredmény is született, melyeket külön kutatási kérdésekben fogalmaztam meg. A fejezet elején röviden ismertetem ezen kérdéseket és azok jelentőségét, majd egymás után az előállt eredmények alapján megválaszolom őket.

4.1. A dolgozat eredményeit összefoglaló kérdések, és azok jelentősége

1. *Milyen a kapcsolat jellege és erőssége az egyes felmérések szubjektív véleményei és a különböző forráskódmetrikák között?*

Az alacsony objektív és magas szintű szubjektív mérőszámok közötti szakadék áthidalásához elengedhetetlen megismerni a köztük levő kapcsolat jellegét és erősségét. A kérdés megválaszolásával feltárható, hogy mely metrikák hogyan és mennyire befolyásolják az egyes altulajdonságokat, így meghatározhatjuk, milyen metrikák elengedhetetlenek egy hatékony modell építéséhez.

2. *Milyen hatékonyan képesek a jól ismert forráskód metrikák alapján az egyes gépi tanulási módszerek megjósolni a metódusokra érkező hallgatói véleményeket?*

A dolgozat legjelentősebb kérdése, hiszen egy hatékony modell segítségével egy több ezer metódusból álló rendszer karbantarthatóságát percek alatt felmérhetjük, rámutathatunk annak gyenge pontjaira. A legrosszabb metódusok pedig alapot szolgáltathatnak egy kódjavító eljárásnak.

3. *Az alkalmazott metódus-véleményezési technikák milyen hatékonyságbeli eltérést mutatnak?*

A gépi tanulás eredményeként előállt modell hatékonysága nagyban függött attól, hogy tanuláshoz használt adatokat miképpen gyűjtöttük. Magasszintű, szubjektív tulajdonságok megismerése során gyakran találkozhatunk az irodalomban különböző felmérésekkel. Azonban a felmérésekben egymástól eltérő rendszerekről, eltérő kérdéseket tesznek fel és ráadásul általában az eredmények előállítása is különböző módon történik. Mivel a felméréseket és az eredmények számolását azonos körülmények között végeztük, így az összehasonlítás az esetünkben teljesen objektív. A kérdés megválaszolásával választ kaphatunk arra, hogyan érdemes a felméréseket összeállítani, mennyi embert és hogyan érdemes megkérdezni, ezáltal hatékonyabb modell építése lehetséges.

4. *Hogyan függ össze a hallgatók egyetemi teljesítménye és a metódusokra adott válaszaik szakértőkhöz viszonyított pontossága?*

Egy kiegészítő fejezete a dolgozatnak, melyben megvizsgáljuk, hogy amennyiben egy hallgató magasabb gyakorlati pontszámot ért el, akkor átlagosan közelebb szavazott-e szakértői véleményhez. A statisztikák számolásához a Rendszerfejlesztés tantárgy pontszámait használtuk, hiszen ezen a kurzuson szerzik a hallgatók a legfontosabb gyakorlati tudást, amely szükséges egy addig ismeretlen Java metódus értékeléséhez.

4.2. Alkalmazott forráskód metrikák és a különböző felmérések véleményeinek kapcsolata

4.2.1. Az alkalmazott metrikák

Mielőtt ismertetném milyen kapcsolat áll fent a különböző metrikák és a szubjektív vélemények között, érdemes közelebbről megvizsgálni az alkalmazott metrikákat és az egymáshoz való viszonyukat. Használt metrikák részletesebb leírása megtalálható a mellékletben.

- *NOI*: Number of Outgoing Invocations - Kimenő hívások száma
- *LOC*: Lines Of Code - Sorok száma
- *LLOC*: Logical Lines Of Code - Nem üres és nem komment sorok száma
- *NOS*: Number Of Statements - Szerkezetek száma
- *NLMA*: Number of Local Methods Accessed - Használt helyi metódusok száma
- *NL*: Nesting Level - Beágyazottsági szint
- *NFMA*: Number of Foreign Methods Accessed - Használt külső metódusok száma
- *NII*: Number of Incoming Invocations - Bejövő hívások száma
- *NPAR*: Number of Parameters - Paraméterek száma
- *McCC*: McCabe Cyclomatic Complexity - McCabe féle ciklomatikus komplexitás
- *CC*: Clone Coverage - Klón lefedettség
- *PMD*: PMD szabálysértések száma [17] a metódusban

2. táblázat. Pearson (R^2) korreláció a metrikák között

	NOI	LOC	LLOC	NOS	NLMA	NL	NFMA	NII	NPAR	McCC	CC	PMD
NOI	1	0.19	0.15	0.12	0.29	0.12	0.81	0.00	0.01	0.05	0.01	0.03
LOC		1	0.98	0.91	0.11	0.34	0.11	0.01	0.01	0.77	0.01	0.69
LLOC			1	0.95	0.11	0.34	0.08	0.01	0.00	0.85	0.01	0.77
NOS				1	0.11	0.32	0.06	0.01	0.00	0.89	0.00	0.85
NLMA					1	0.08	0.01	0.01	0.00	0.10	0.01	0.08
NL						1	0.07	0.01	0.01	0.31	0.01	0.27
NFMA							1	0.01	0.01	0.01	0.02	0.01
NII								1	0.00	0.01	0.00	0.01
NUMPAR									1	0.00	0.00	0.00
McCC										1	0.00	0.93
CC											1	0.00
PMD												1

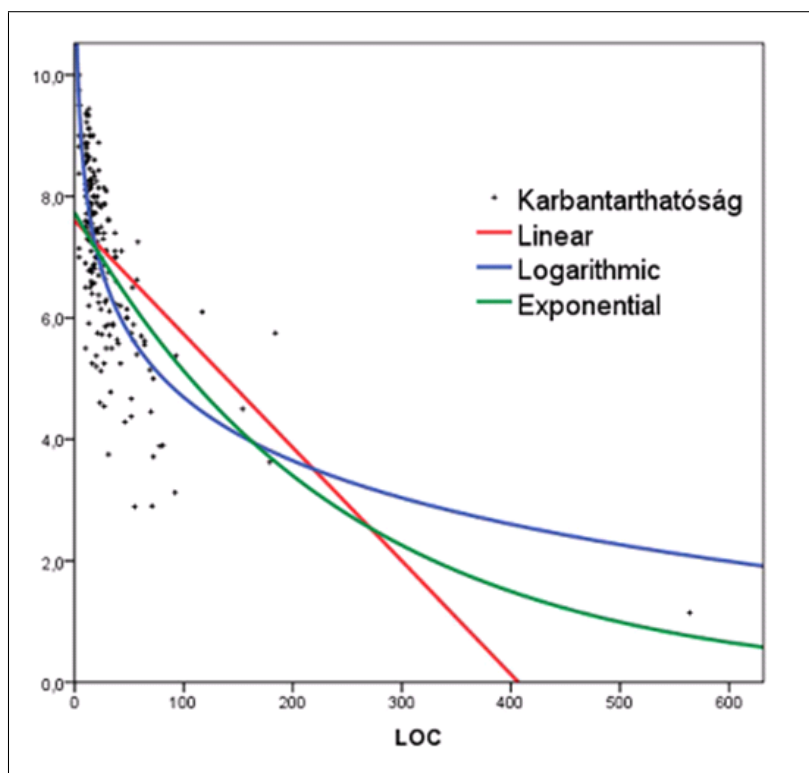
Korábban már számos tanulmány foglalkozott a metrikák közötti kapcsolatok feltárásával, azonban ez legtöbbször osztály szinten történt, mi most a metódusok szintjén végezzük el az összehasonlítást. A 2. táblázat tartalmazza a jEdit rendszer metódusaihoz tartozó metrikaértékek közötti Pearson (R^2) korrelációt. Nagyon magas korrelációt tapasztaltunk LOC, LLOC, NOS, McCC és PMD metrikaértékek között. Nem meglepő, hogy a méret szintű metrikák (LOS, LLOC, NOS) jól korrelálnak. Azonban a nagyon magas 0.93-mas korreláció az McCC és PMD között igen érdekes, hiszen ez azt jelenti, hogy minél komplexebb egy metódus, annál több PMD szabálysértést tartalmaz.

4.2.2. Kapcsolat erősségének összehasonlítása regressziós eljárások szerint

Amikor két mennyiség kapcsolatát vizsgáljuk legtöbbször Pearson korrelációs együtthatót számolunk, ami azonban lineáris kapcsolatot feltételez a mennyiségek között, pedig előfordulhat más típusú is, például logaritmikus vagy exponenciális is. Az SPSS-ben¹ [19] elérhető görbeillesztési regressziós eljárások futtatása után azt tapasztaltam, hogy a lineáris, logaritmikus és exponenciális görbék illeszkednek leginkább a pontokra. Ezeket a görbéket a hallgatói kiértékelés karbantarthatóság tulajdonsága és a LOC metrika között ábrázoltam az 5. ábrán. A lineáris görbe 0.326, az exponenciális 0.512 a logaritmikus pedig 0.561-es R^2 korrelációs értékkel közelíti a pontokat, a táblázatban ezen értékek négyzetgyökei szerepelnek.

A három regresszió közül a lineáris és logaritmikus regresszió által biztosított korrelációs értékeket fogom összehasonlítani. Logaritmikus jellegű kapcsolatot úgy értelmeztem, hogy vettem a metrikák logaritmusát és így vizsgáltam a két mennyiség közötti

¹ Az SPSS(Statistical Package for the Social Sciences) egy széleskörűen használt szoftver adatok gyűjtésére, adatbányászatra és statisztikai elemzések elvégzésére.



5. ábra. Karbantarthatóság és a LOC metrika lineáris, logaritmus és exponenciális kapcsolatának összehasonlítása

Pearson korrelációt. Ez a módszer az SPSS-ben számolt logaritmus regresszióval megegyező korrelációs értéket adott.

A következőkben megnézzük, a kapott korrelációs értékeket részletesen a hallgatói kiértékelés esetén.

3. táblázat. Lineáris kapcsolat erőssége a forráskódmétrikák és a hallgatói vélemények átlaga között

	NOI	LOC	LLOC	NOS	NLMA	NL	NFMA	NII	NPAR	McCC	CC	PMD
Elemezhetőség	-0.51	-0.54	-0.50	-0.47	-0.35	-0.41	-0.43	0.00	-0.08	-0.37	0.07	-0.38
Változtathatóság	-0.54	-0.50	-0.45	-0.42	-0.36	-0.38	-0.47	-0.03	-0.06	-0.33	0.07	-0.32
Stabilitás	-0.49	-0.46	-0.41	-0.38	-0.28	-0.35	-0.45	0.00	-0.15	-0.29	0.05	-0.30
Tesztelhetőség	-0.47	-0.58	-0.55	-0.51	-0.31	-0.17	-0.41	-0.05	0.00	-0.39	0.06	-0.13
Érthetőség	-0.41	-0.47	-0.43	-0.40	-0.29	-0.39	-0.34	-0.06	-0.06	-0.32	0.05	-0.32
Karbantarthatóság	-0.57	-0.57	-0.52	-0.48	-0.38	-0.43	-0.49	-0.02	-0.14	-0.37	0.10	-0.37

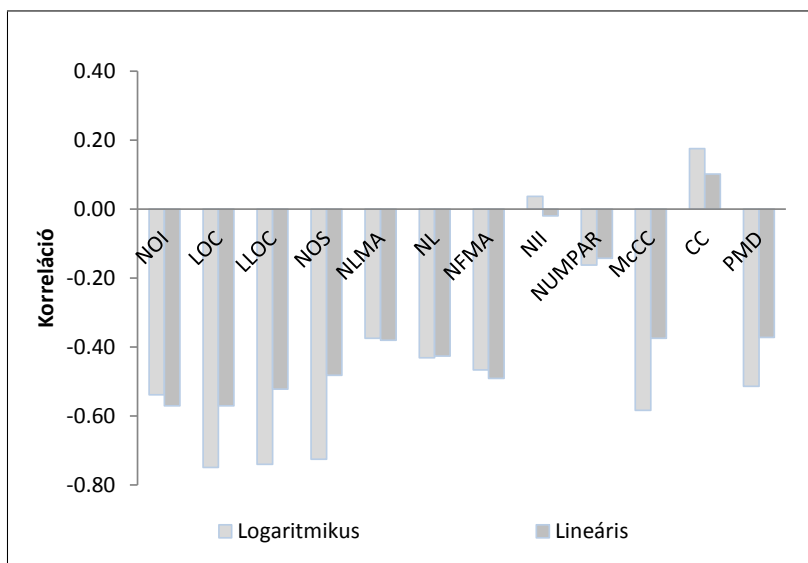
A 3. és 4. táblázatok tartalmazzák a hallgatói kiértékelések által kapott válaszok átlagának és a metrikák közötti lineáris és logaritmus regresszió által elért korrelációs értékeket. Az elvártaknak megfelelően szinte minden érték negatív, vagyis minél nagyobb

4. táblázat. Logaritmikus kapcsolat erőssége a forráskódmetriák és a hallgatói vélemények átlaga között

	NOI	LOC	LLOC	NOS	NLMA	NL	NFMA	NII	NPAR	McCC	CC	PMD
Elemezhetőség	-0.49	-0.70	-0.69	-0.69	-0.34	-0.40	-0.42	0.05	-0.12	-0.54	0.13	-0.49
Változtathatóság	-0.53	-0.66	-0.64	-0.64	-0.36	-0.38	-0.46	0.04	-0.10	-0.52	0.17	-0.43
Stabilitás	-0.50	-0.67	-0.65	-0.64	-0.28	-0.36	-0.45	0.03	-0.20	-0.50	0.10	-0.45
Tesztelhetőség	-0.34	-0.62	-0.63	-0.57	-0.26	-0.32	-0.32	-0.09	-0.07	-0.52	0.26	-0.33
Érthetőség	-0.40	-0.62	-0.62	-0.60	-0.28	-0.40	-0.32	-0.03	-0.09	-0.53	0.17	-0.43
Karbantarthatóság	-0.54	-0.75	-0.74	-0.73	-0.37	-0.43	-0.47	0.04	-0.16	-0.58	0.18	-0.51

egy metrika, annál rosszabb lesz az adott tulajdonság. Az NII, NUNPAR, CC metrikák kivételével a metrikák közepes/erős kapcsolatot mutatnak minden tulajdonsággal. Megfigyelhető, hogy a számos metrika esetben a logaritmikus regresszió által adott korrelációs értékek jobbak, mint a lineáris regresszió esetén.

A 6. ábráról leolvasható a metrikák és a karbantarthatóság közötti lineáris és logaritmikus kapcsolat erőssége. Megállapítható, hogy az McCC, a PMD és a méret alapú metrikák esetében sokkal jobb eredményeket kaptunk.

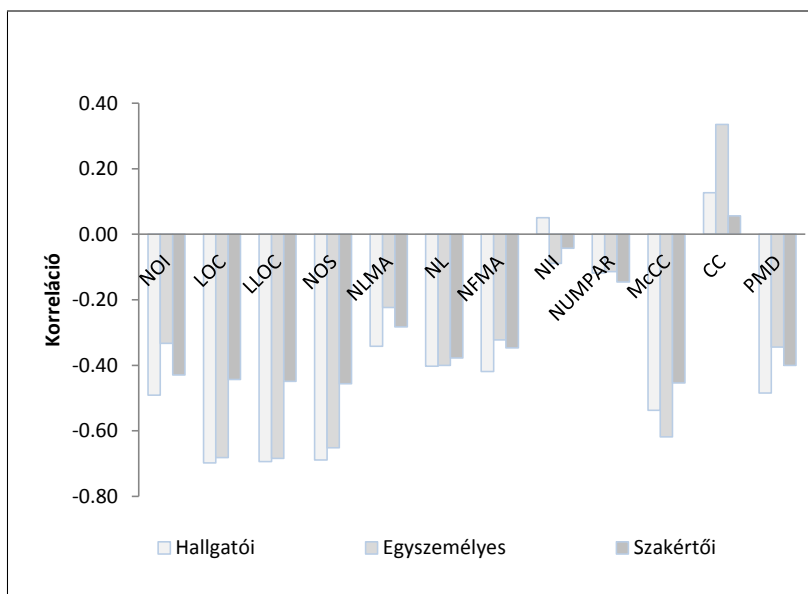


6. ábra. Lineáris és Logaritmikus kapcsolat összehasonlítása a hallgatói karbantarthatósági válaszok alapján

4.2.3. Kapcsolat erősségének összehasonlítása felmérések szerint

A felmérés szintű adatok részletesen megtalálhatóak a mellékletben, azonban ebben a fejezetben csak az egyik tulajdonság elemzésére van lehetőség. Ennek megfelelően ki-

választottam a felmérés szintű összehasonlításhoz az első tulajdonságot, az elemezhetőséget. A 7. ábráról és a mellékletben található táblázatokból leolvasható, hogy a legtöbb esetben a szakértői kiértékelés mutat a legkisebb kapcsolatot a metrikákkal. Lineáris kapcsolat erőssége hasonló az egyszemélyes és hallgatói felmérések során. A logaritmikus kapcsolat viszont erősebb a hallgatói kiértékelés esetében.



7. ábra. Metrikák és az egyes felmérések által adott elemezhetőségi vélemények közötti logaritmikus regresszió korrelációs értéke

Elegendő információ áll rendelkezésünkre, hogy megválaszolhassuk az első kutatási kérdést.

1. kérdés: *Milyen a kapcsolat jellege és erőssége az egyes felmérések szubjektív véleményei és a különböző forráskódmetrikák között?*

A fenti és a mellékletben szereplő eredmények alapján megállapítható, hogy a méret, komplexitás, PMD szabálysértés alapú metrikák sokkal inkább logaritmikus, mint lineáris kapcsolatban vannak az egyes tulajdonságokkal. A többi metrika esetén hasonló a kapcsolat a két esetben. Az egyes felmérések esetén a szakértői kiértékelés mutat a legkisebb kapcsolatot a metrikákkal, hiszen eltérő mértékben vették őket figyelembe a kiértékelés során a szakértők. A hallgatói kiértékelés véleményei logaritmikus esetben jobb, lineáris esetben hasonlóan erős kapcsolatot mutatnak a metrikákkal, mint az egyszemélyes kiértékelő véleményei. Ez arra enged következtetni, hogy a hallgatói vélemények átlaga egy globális véleményhez közelít, amely vélemény olyan kapcsolatot mutat a metrikákkal, mintha egy ember értékelt volna ki őket.

4.3. Gépi tanulási módszerek hatékonysága a különböző felmérések esetén

4.3.1. Alkalmazott gépi tanulási módszerek

A gépi tanulás egy olyan változatos problémák megoldására szolgáló általános módszer, melynek célja a múltbéli esetek felhasználásával olyan modellt építeni, amely képes addig még nem látott esetek értékének a közelítésére. Egy ilyen igazoltan jól működő modell akár képes lehet helyettesíteni különböző területek szakértőinek véleményét is. Például megállapíthatja egy szövegről, hogy az rákos vagy sem, érdemes-e jobban megvizsgálni azt. A mi esetünkben egy olyan gépi tanulási modell építése a cél, amely hatékonyan képes közelíteni egy metódus karbantarthatóságát. A metódusok karbantarthatóságának ilyen jellegű automatikus meghatározására azért van szükség, hiszen egy nagyobb méretű rendszer akár több tízezer metódusból is állhat, melyek egyenként történő vizsgálatára nincs lehetőség.

Ahhoz, hogy a modell megválaszolhasson ilyen bonyolult problémákat szükséges egy automatikusan gyorsan kinyerhető tulajdonságvektor meghatározása minden esetre. Ezt a tulajdonságvektort a mi esetünkben a metódus szintű forráskódmetrikák jelentik. A választok alapján a gépi tanuló algoritmus modellt épít, amely egy konkrét metrikasorozathoz hozzárendel egy karbantarthatósági tulajdonság értékét, amely a reményeink szerint tükrözi a tényleges véleményt. A modell felépítéséhez szükséges metódus-kiértékeléseket felmérések segítségével gyűjtöttük, melyek során több száz metódus ISO/IEC 9126 szabványban definiált karbantarthatósághoz tartozó altulajdonságokról kérdeztünk meg tapasztalt szoftverfejlesztőket és a Szegedi Tudományegyetem informatikus hallgatóit. A modell eredményességét 10-szeres keresztvalidáció segítségével határoztuk meg, amely során az adatok 90%-n a rendszer tanítja a modellt és a maradék 10%-on pedig teszteli annak eredményességét. Ezt elvégezve 10-szer egymás után kiszámolható, hogy átlagosan mennyire hatékonyan képes a modell egy metódusnak megmondani az egyes tulajdonságait.

A gépi tanulás klasszikus formája az osztályozás, mely során az osztályozandó változó típusa nominális, vagyis lehetséges értékeinek a halmaza véges és köztük nem állítható fel semmilyen arányszám, sorrend. A metódusok osztályozására a [16] cikkben a jó, közepes, rossz osztályozást használtuk. Az ilyen típusú tanulásnak a legfőbb mérőszáma a helyesen osztályozott egyedek aránya. A probléma ebben a megközelítésben az volt, hogy az esetek nagy része (kb. 60%-a) a jó osztályba tartozott és modell ezt úgy kezelte, hogy a legtöbb metódust jó osztályba sorolta és alacsony lett a rossz metódusok megtalálásának aránya (recall érték), ami az igazi célunk lett volna.

A másik lehetőség modell építésére a regresszió használata, mely során a változó értéke valós szám típusú, és a modell is hasonló típusú közelítést ad. A regressziós modell hatékonyságára a tényleges és a modell által szolgáltatott számok közötti Pearson-korreláció, valamint az átlagos abszolút hiba a szokásos mérőszám. Ezt az utóbbi megközelítést használtuk, hiszen számos előnye van. Legnagyobb előnye, hogy sokkal sűrűbb skálát használunk, így pontosabb eredményeket kapunk. A korreláció megmondja mennyire találja el a modell a tendenciát, míg az abszolút relatív hiba jelzi, hogy átlagosan mennyit téved, ezek sokkal többetmondó mérőszámok a hatékonyságra, mint amelyek osztályozás esetén adottak. Továbbá a tanulás által felépített modellt megvizsgálva olyan számszerűsített következtetéseket tudunk levonni, mint például magasabb komplexitáshoz alacsonyabb tesztelhetőség tartozik.

A Metrika Kiértékelő Keretrendszer a felmérések során gyűjtött válaszokat és metrikákat összegyűjti tulajdonságokként egy-egy fájlba. Ezen fájlok és a Weka [8] adatbányászati eszköz segítségével könnyedén elvégezhettem számos gépi tanulási algoritmus hatékonyságának tesztelését.

4.3.2. Különböző algoritmusok hatékonyságának összehasonlítása

Először a hallgatói kiértékelés segítségével megvizsgálom az egyes gépi tanulási algoritmusok hatékonyságát, majd összehasonlítást végzek az egyes felmérések között. A vizsgált tanuló algoritmusok között szerepelt döntési fa, neurális háló és többváltozós lineáris regresszió is. Lineáris regresszió és a neurális háló esetén a Weka egyetlen opciót kínált, azonban a döntési fa esetén kiválasztottam a legjobbat. Ezt a REPTree döntési fa algoritmust egy bagging nevű technikával tovább tudtam javítani, amely a tanuló halmazon nem csak egy fát, hanem többet is felépít, majd ezek közül egy olyat választ, ami általánosan jobban teljesít a teszteseteken is. Az előállt eredmények hatékonyságát a korrelációs és hiba mutatók mellett jelzi a ZeroR nevű algoritmus is, amely predikcióként mindig a tanulóhalmazban levő osztályozandó változó értékeinek az átlagát választja. A metrikák használata nélkül ez a választás adja a legkisebb átlagos hibával járó predikciót. Így össze lehet hasonlítani, hogy a metrikák használata mennyiben javít az eredményeken.

Az 5. és 6. táblázatokban összegyűjtöttem, hogy a hallgatói kiértékelés altulajdonságait milyen korrelációval és átlagos hibával voltak képesek megjósolni az egyes algoritmusok. A döntési fa algoritmus nagyságrendekkel jobban teljesített, mint a neurális háló, illetve a többváltozós lineáris regresszió, hiszen sokkal nagyobb átlagos korrelációval (0.631) és sokkal kisebb átlagos hibával (0.87) képes predikálni az egyes tulajdonságokat. A következőkben a felmérések szintjén összehasonlítom ezt a döntési fa algoritmust, majd megnézzük mi történik, ha a metrikák logaritmusát használjuk a tanuláshoz.

5. táblázat. Különböző gépi tanulási módszerek korrelációja a hallgatói kiértékelés során

	ZeroR	Neurális Háló	Lin. Regresszió	Döntési fa
Elemezhetőség	-0.162	0.408	0.466	0.660
Változtathatóság	-0.116	0.362	0.437	0.571
Érhetőség	-0.153	0.275	0.491	0.621
Stabilitás	-0.239	0.297	0.360	0.572
Tesztelhetőség	-0.118	0.427	0.363	0.639
Karbantarthatóság	-0.122	0.587	0.615	0.723
Átlag	-0.152	0.393	0.455	0.631

6. táblázat. Különböző gépi tanulási módszerek átlagos hibája a hallgatói kiértékelés során

	ZeroR	Neurális Háló	Lin. Regresszió	Döntési fa
Elemezhetőség	1.201	1.076	1.076	0.884
Változtathatóság	1.026	1.088	0.965	0.861
Érhetőség	1.574	1.387	1.188	1.048
Stabilitás	0.822	0.824	0.833	0.670
Tesztelhetőség	1.189	1.168	1.145	0.926
Karbantarthatóság	1.187	1.193	0.909	0.831
Átlag	1.166	1.123	1.019	0.870

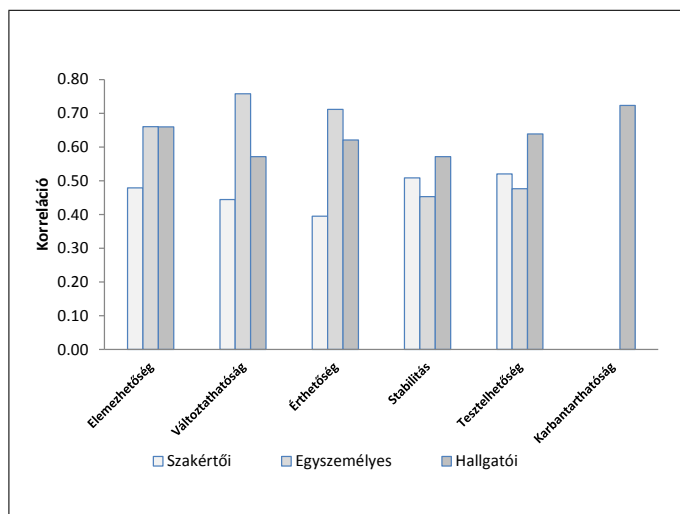
4.3.3. Felmérések szerinti összehasonlítás

7. táblázat. Döntési fa hatékonysága az egyes felmérések esetén

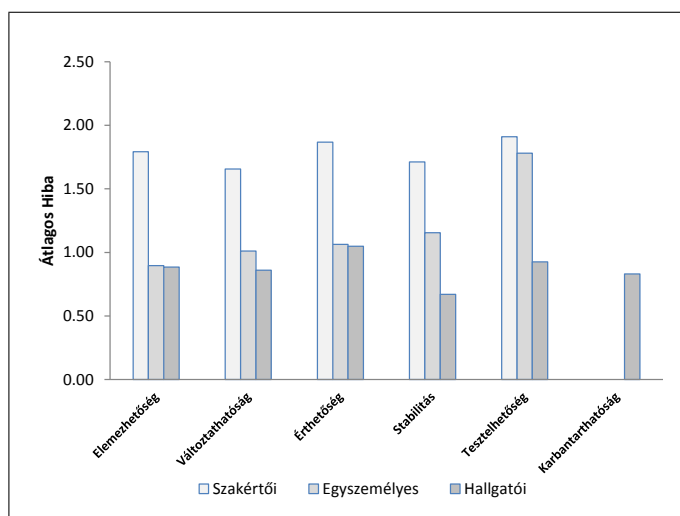
	Szakértői		Egyszemélyes		Hallgatói	
	Átlagos hiba	Korreláció	Átlagos hiba	Korreláció	Átlagos hiba	Korreláció
Elemezhetőség	1.792	0.479	0.896	0.660	0.884	0.660
Változtathatóság	1.656	0.445	1.011	0.758	0.861	0.571
Stabilitás	1.712	0.509	1.154	0.453	0.670	0.572
Tesztelhetőség	1.910	0.520	1.781	0.476	0.926	0.639
Érthetőség	1.867	0.395	1.063	0.712	1.048	0.621
Átlag	1.787	0.469	1.181	0.612	0.878	0.612

Korábban megnéztük, hogy a döntési fa algoritmus nagyságrendekkel jobban teljesít, bármelyik másik algoritmusnál, éppen ezért a felmérések szintjén elegendő ennek az algoritmusnak a megvizsgálása. A 7. táblázatban összegyűjtöttem az egyes felmérésekhez tartozó mutatókat. Valamint a 8. és 9. ábrákon ábrázoltam is az értékeket. A hallgatói kiértékelés során értékeltük a karbantarthatósági tulajdonságok mellett magát a karbantarthatóság tulajdonságot is. Azonban ez az adat, így csak a hallgatói felmérés esetén állt rendelkezésünkre, ezért az ábrákon csak egy oszlop tartozik hozzá.

Az átlagos korrelációs (0,469) és átlagos hiba (1,787) értékekből leolvasható, hogy a szakértői kiértékelés esetén a döntési fa nem tudott elég hatékony modellt felépíteni. Ugyanakkor igen érdekes, hogy a szakértői modell alapján két legjobban predikált tulajdonság a stabilitás és tesztelhetőség. Ez a két tulajdonság pontosan megegyezik az egyszemélyes modell által legrosszabbul predikált tulajdonságokkal. Emiatt az érdekes egybeesés miatt a szakértői modell jobban predikálja (korreláció szerint) ezt a két tulajdonságot, mint az egyszemélyes modell. Az egyszemélyes és a hallgatói predikciók átlagos korrelációs értéke szintén igen érdekes módon pontosan megegyezik, ugyanakkor a hallgatói kiértékelésnél az átlagos hiba sokkal kisebb.



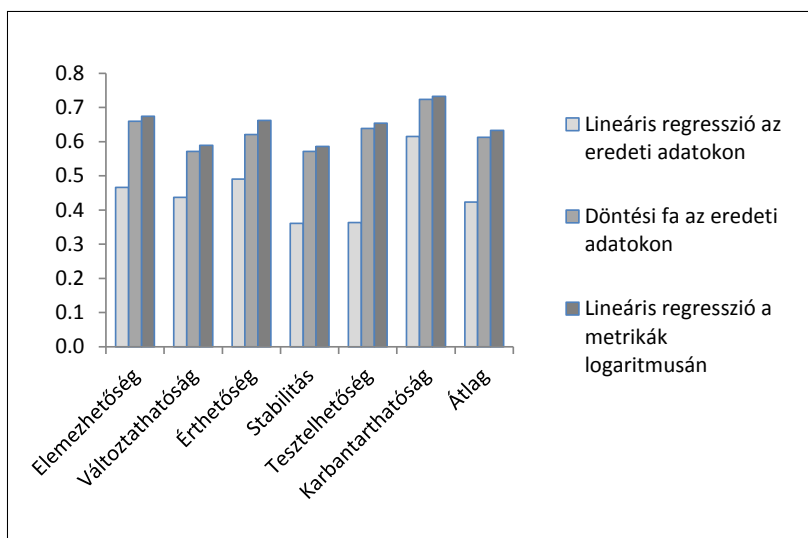
8. ábra. Döntési fa által megjósolt és a tényleges adatok közötti Pearson (R) korreláció



9. ábra. Döntési fa algoritmus relatív abszolút hibája

4.3.4. Többváltozós lineáris regresszió a metrikák logaritmusán

A metrikák és vélemények közötti kapcsolatról a korábbi fejezetben beláttuk, hogy sokkal inkább logaritmikus, mint lineáris, ezek után adódik a gondolat, hogy miért nem használjuk a predikcióhoz a metrikák logaritmusát, ezzel segítve a tanuló algoritmusok feladatát. A kísérlet elég érdekes eredményeket szült, a döntési fa algoritmus hatékonysága kicsit romlott, ugyanakkor a többváltozós lineáris regresszió nagyon sokat javult. A 10. ábrán látható módon a hallgatói kiértékelés során jelentősen jobb eredményeket ért el, mint a döntési fa az eredeti adatokon, azonban a másik kettő felmérés esetben maradt a sorrend.

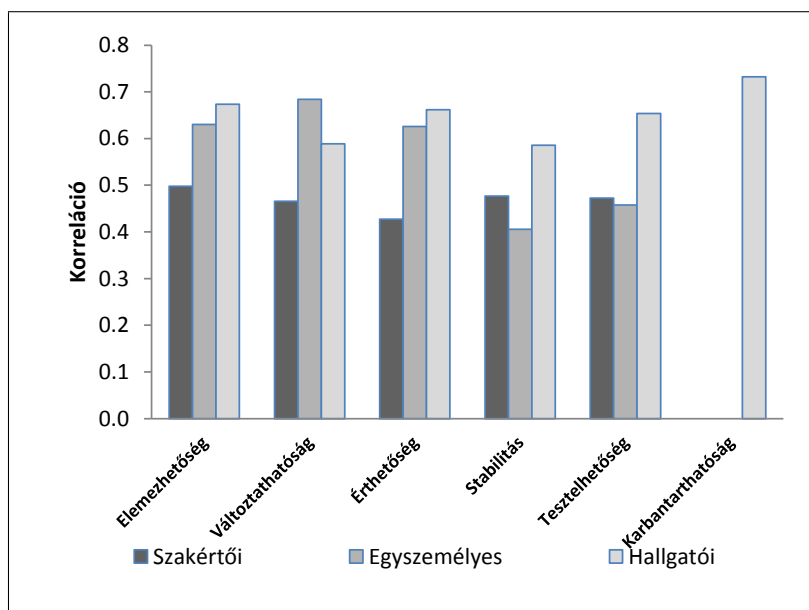


10. ábra. Lineáris regresszió és Döntési fa korrelációk a hallgatói kiértékelés adatain

A fenti ábra mindenképpen alátámasztja, hogy mekkora jelentősége van a metrikák és a tulajdonságok közötti kapcsolat tanulmányozásának, hiszen az eredeti adatokon tanult legszofisztikáltabb algoritmusoknál az adatok logaritmusán tanított egyszerű többváltozós lineáris regresszió jelentősen jobban teljesített az összes tulajdonság esetén.

8. táblázat. Többváltozós lineáris regresszió hatékonysága az egyes felmérések esetén

	Szakértői		Egyszemélyes		Hallgatói	
	Átlagos hiba	Korreláció	Átlagos hiba	Korreláció	Átlagos hiba	Korreláció
Elemezhetőség	1.767	0.498	1.075	0.630	0.860	0.674
Változtathatóság	1.640	0.466	1.266	0.684	0.849	0.589
Stabilitás	1.767	0.477	1.164	0.406	0.666	0.586
Tesztelhetőség	1.922	0.473	1.783	0.458	0.902	0.654
Érthetőség	1.826	0.427	1.168	0.626	0.978	0.662
Átlag	1.784	0.468	1.291	0.561	0.851	0.633

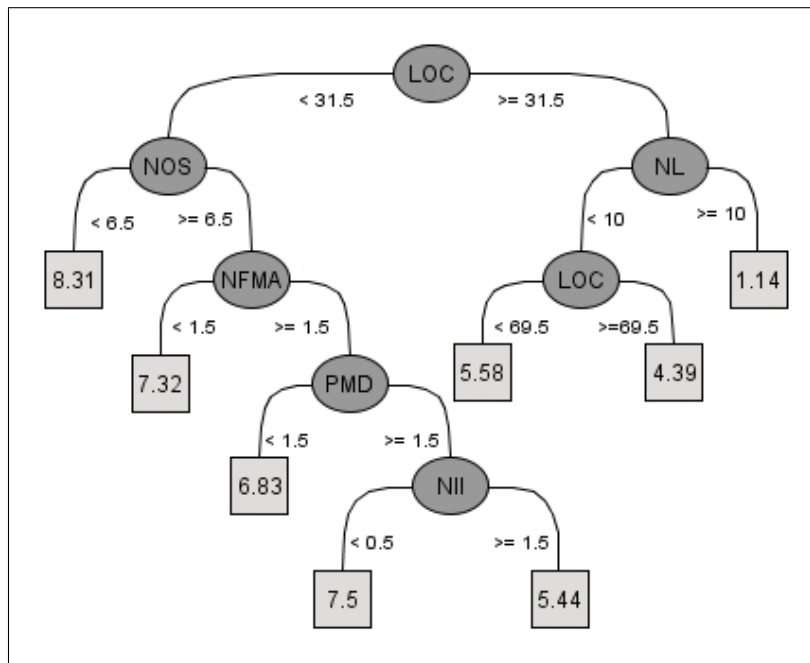


11. ábra. Többváltozós Lineáris regresszió által megjósolt és a tényleges adatok közötti Pearson (R) korreláció

4.3.5. Modellek elemzése

Ebben az alfejezetben röviden bemutatok három példát a fenti eredmények alapján hatékonynak ítélt döntési fa és lineáris regressziós modellek közül. Megvizsgáljuk, hogy a modellek nem csak az elvárt eredményeket adják, hanem az elvártaknak megfelelőképpen épülnek-e fel.

Az első modell a bevezetőben is már említett 0.72-es korrelációval és 0.83-mas abszolút relatív hibával képes megbecsülni egy [0,10]-es folytonos skálán a hallgatói vélemények átlagát a karbantarthatóságról, ahol a 0 jelenti a nagyon rosszul karbantartható metódust, míg a 10 a tökéletesen karbantarthatót. Az ilyen módon előállt modell megtalálható a 12. ábrán. Leolvasható, hogy a legfelső szinten a méret alapú metrikák (LOC, NOS) szerepelnek, majd a különböző komplexitás (NL), szabálysértés (PMD) és további metrikák (NFMA, NII) pontosítják a modellt. Megállapítható tehát, hogy a modell változatos típusú metrikákat használ, ami arra utal, hogy a hallgatók figyelembe vették nem csak metódus méretét, hanem például annak beágyazottsági szintjét, szabálysértéseinek számát is.



12. ábra. Hallgatói kiértékelés Karbantarthatóság tulajdonságának becslésére felépített döntési fa

Az egyváltozós lineáris regresszió megkeresi azt az $ax + b$ egyenest, amely legjobban közelíti a tanulóhalmaz értékeit ezzel összhangban a többváltozós eset is hasonlóan működik csak több dimenzióban keresi az $a_1x_1 + a_2x_2 + \dots + a_nx_n + b$ hipersíkot.

Megvizsgáljuk, hogy a lineáris regresszió a metrikák logaritmusan a fenti döntési fához képest, hogyan épít fel egy hatékonyabb modellt a karbantarthatóság becslésére. A korreláció ebben az esetben 0.73, az átlagos hiba pedig 0.81. A modell szintén méret alapú metrikákat használ a legnagyobb súllyal, majd az így kapott értéket pontosítja egyéb metrikákkal. Érdekes, hogy a döntési fa és a lineáris regressziós modellben is az NII metrika növeli a tulajdonság értékét, tehát minél több helyen hívnak meg egy metódust, annál karbantarthatóbb. Érdeemes megjegyezni, hogy a modellben szereplő értékek a metrikák logaritmikus értékeire vonatkoznak.

Karbantarthatóság =

$$\begin{aligned}
 &-1.1779 * LOC + \\
 &0.3028 * LLOC + \\
 &-0.3438 * NOS + \\
 &-0.4395 * NLMA + \\
 &-0.1895 * NFMA + \\
 &0.2571 * NII + \\
 &-0.1953 * NUMPAR + \\
 &-0.1423 * McCC + 11.0988
 \end{aligned}$$

Közismert, hogy bonyolult és sok paraméterrel rendelkező metódusok tesztelése nehézkes, ennek megfelelően azt várjuk, hogy ezek negatívan befolyásolják a modellben a tesztelhetőséget. A tesztelhetőség becslésekor mindössze négy metrikát használt a modell, ezekből a három méret alapú egymással nagyon erősen korrelál ($R^2 > 0.9$), így például a LLOC pozitív hatását, a NOS és LOC kioltja. A negyedik tőlük teljesen független ($R^2 \leq 0.01$) metrika a NUMPAR, vagyis a paraméterek száma. Ki is számolhatjuk, hogy a metrika egy egységnyi logaritmikus változása -0.4266-al csökkenti a tesztelhetőséget a modell szerint.

Tesztelhetőség =

$$\begin{aligned} & -1.8082 * LOC + \\ & 0.9761 * LLOC + \\ & -0.3398 * NOS + \\ & -0.4266 * NUMPAR + 10.3094 \end{aligned}$$

4.3.6. Eredmények összefoglalása

A fenti összesítő táblázatok alapján megválaszolhatunk a kutatási kérdések közül kettőt is, hiszen megnéztük a közismert gépi tanuló algoritmusok hatékonyságát, és össze is hasonlítottuk őket a felmérések szintjén.

2. kérdés: *Milyen hatékonyan képesek a jól ismert forráskód metrikák alapján az egyes gépi tanulási módszerek megjósolni a metódusokra érkező véleményeket?*

Kijelenthető, hogy a neurális háló nem képes hatékonyan predikálni a véleményeket, míg a döntési fa megfelelő felmérés esetén egyenletesen hatékonyan képes. A lineáris regresszió pedig nagyon hatékonyan működik a metrikák logaritmusán, de csak a hallgatói kiértékelés során. Mivel a REPTree döntési fa algoritmus egyenletesen jól teljesített minden felmérés során, így általánosan jó módszernek ezt lehet tekinteni. A hallgatói karbantarthatósági véleményeket például 0,72-es korrelációval és 0,83-as abszolút hibával közelítette. Ez mindenképpen egy erős korrelációs értéknek tekinthető.

3. kérdés: *Az alkalmazott metódus-véleményezési technikák milyen hatékonyságbeli eltérést mutatnak?*

A felmérések szintjén összehasonlítottuk a döntési fa algoritmus hatékonyságát, illetve a metrikák logaritmusán tanított többváltozós regressziós algoritmus hatékonyságát. Arra a következtetésre juthattunk, hogy a hallgatói kiértékelés során egyenletesen hatékonyan, míg az egyszemélyes kiértékelés során a stabilitás és tesztelhetőség kivételével hatékonyan lehet a szubjektív véleményeket megjósolni. A szakértői felmérés véleményeit nem tudtuk elég hatékonyan predikálni.

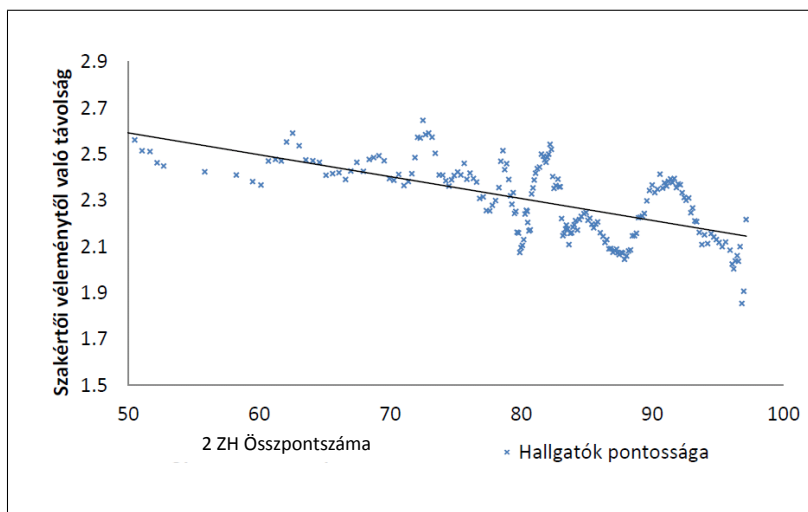
4.4. Hallgatók teljesítménye és a szakértői véleményekhez viszonyított pontatlanságának kapcsolata

A hallgatói felmérés során szándékosan ugyanazokat a metódusokat értékeltük ki, mint amelyeket korábban a szakértői kiértékelés során. Tehát minden hallgató kiértékelte metódusai mellett szerepel egy szakértői vélemény is. Így ki lehet számolni, hogy átlagosan mekkora különbség figyelhető meg a szakértői és adott hallgató véleményei között minden tulajdonság esetén. A tulajdonságonként számolt pontatlanságok átlagaként megkapjuk, hogy egy hallgató átlagosan milyen messze szavazott a szakértői véleményétől.

Mivel a hallgatók a Szegedi Tudományegyetemen egy haladó programozási kurzus keretei (Rendszerfejlesztés) között végezték a kiértékelést, ezért minden hallgatóra adott volt a kurzussal kapcsolatos teljesítménymutatói is (ZH pontszám, kollokvium pontszám, kollokvium jegy).

A 13. ábra azon tendenciára mutat rá, mely szerint a teljesítmény növekedésével (ZH pontszám) javul a hallgató szavazatainak pontossága. Tehát egyre inkább szakértőként értékeli a metódusokat ő maga is. A két mennyiség között egy simítás után a Pearson korrelációs érték -0.703 . Ezen statisztika segítségével választ is adhatunk az utolsó, negyedik kérdésre.

4. kérdés: *Hogyan függ össze a hallgatók egyetemi teljesítménye és a metódusokra adott válaszaik szakértőkhöz viszonyított pontossága?* A fenti adatok alapján tehát, ahogyan nő a hallgató ZH pontszáma -0.703 -as korrelációval csökken a szakértőkhöz viszonyított pontatlansága. Vagyis minél jobban teljesít egy hallgató Rendszerfejlesztés tárgyból, annál inkább közelít a metódusokról alkotott véleménye egy szakértő véleményéhez.



13. ábra. Hallgatók szavazatainak szakértőkhöz viszonyított pontossága a ZH pontszámuk függvényében

5. Összefoglalás, jövőbeli elképzelések

A dolgozatnak alapvető feladata a metrikák és az ISO/IEC 9126-os szabványban szereplő karbantarthatósági tulajdonságok közötti kapcsolat megismerése, majd a metrikák segítségével a tulajdonságok hatékony predikálása volt. Az elért eredményeket mindig összehasonlítottam a különböző felmérések szintjén is, ezáltal képet kaphattunk azok előnyeiről, hátrányairól. A metrikák és a tulajdonságok közötti kapcsolatról megállapítottam, hogy a természetesnek vett lineáris kapcsolat helyett sokkal inkább a logaritmikus jellemző a legtöbb metrikára, tehát ahogy nő a metrika úgy lesz logaritmikus ütemben egyre kisebb a tulajdonság. Ebből az eredményből született a gondolat, miszerint ha vesszük a metrikák logaritmusát, azzal megkönnyíthetjük a gépi tanulási módszerek feladatát. A hallgatói kiértékelés során így hatékonyabb modellt sikerült felépíteni, mint az eredeti metrikaértékek alapján.

A szakértői, egyszemélyes és hallgatói felmérések összehasonlítása során megállapítottam, hogy a szakértői kiértékelésből származó véleményeket nem sikerült hatékonyan becsülni a tulajdonságok megítélésében mutatkozó szórás miatt. Ezzel szemben az egyszemélyes és hallgatói kiértékelések véleményei a korrelációt tekintve azonosan hatékonyan megjósolhatóak, azonban a hallgatói predikciók esetén sokkal kisebb az átlagos hiba.

A hallgatói felmérés során sikerült kimutatni a tendenciát, miszerint minél jobb egy hallgató gyakorlati pontszáma, annál inkább közelebb van a véleménye a szakértői véleményhez.

A jól ismert gépi tanulási algoritmusokat összehasonlítva arra a következtetésre jutottam, hogy a legjobban az eredeti metrikákon a döntési fa algoritmus teljesít. A hallgatói kiértékelés során 0.612-es átlagos korrelációval és 0.878-as átlagos hibával közelíti a tulajdonságokat [0,10]-es folytonos skálán, a karbantarthatóság tulajdonságot pedig 0.72-es korrelációval és 0.83-as átlagos hibával. Ugyanakkor a metrikák logaritmusán tanulva lineáris regressziót alkalmazva átlagosan 0.633-as korrelációval és 0.851-es átlagos hibával becsültük a hallgatók karbantarthatósági tulajdonságokról alkotott véleményeit. Ezzel a módszerrel a karbantarthatóság tulajdonságot 0.73-mas korrelációval és 0.81-es átlagos hibával közelítettük.

Ezen értékek alapján megállapítható, hogy hatékonyan tudjuk becsülni a metódusok karbantarthatósági tulajdonságára érkező hallgatói vélemények átlagát.

Láttuk, hogy a metrikák és a tulajdonságok kapcsolatának vizsgálata segítségével a metrikákat olyan formára tudtuk alakítani, amelyen hatékony modell építése volt lehetséges. Elképzelhető, hogy ha minden metrikára külön vizsgáljuk a kapcsolat jellegét, és

egyenként kezeljük őket, akkor jobb eredményeket érhetünk el. Összetettebb, vagy más típusú felmérések kidolgozásával esetleg a már meglévő felmérések egyéb kérdéseire adott válaszok alapján elképzelhető hatékonyabb modell felállítása, vagy a metrikák és a tulajdonságok közötti kapcsolat jobb megismerése. Amennyiben elfogadjuk, hogy hatékonyan tudjuk becsülni ezeket a tulajdonságokat, számos lehetőség adódik, például a rendszer metódusainak rendszer szinten, költségek tekintetében vagy időben történő vizsgálatára.

6. Köszönetnyilvánítás

Ezúton szeretném megköszönni Hegedűs Péter és Dr. Ferenc Rudolf témavezetőimnek a folyamatos támogatásukat és szakértelmüket. A rendszeres konzultációk nagyban hozzájárultak a dolgozat időben történő elkészüléséhez. Köszönet jár továbbá családomnak és páromnak, amiért elolvasták az elkészült dolgozatot.

Hivatkozások

- [1] Automotive SPICE. <http://www.automotivespice.com/>,
<http://www.automotivespice.com/>
- [2] Bakota, T., Hegedűs, P., Körtvélyesi, P., Rudolf, F., Gyimóthy, T.: A Probabilistic Software Quality Model. In: Proceedings of the 27th IEEE International Conference on Software Maintenance. pp. 368–377. ICSM 2011, IEEE Computer Society, Williamsburg, VA, USA (2011)
- [3] Bansiya, J., Davis, C.: A Hierarchical Model for Object-Oriented Design Quality Assessment. IEEE Transactions on Software Engineering 28, 4–17 (2002)
- [4] Basili, V.R., Briand, L.C., Melo, W.L.: A Validation of Object-Oriented Design Metrics as Quality Indicators. In: IEEE Transactions on Software Engineering. vol. 22, pp. 751–761 (Oct 1996)
- [5] Chrissis, M.B., Konrad, M., Shrum, S.: CMMI Guidelines for Process Integration and Product Improvement. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (2003)
- [6] Ferenc, R., Beszédes, Á., Tarkiainen, M., Gyimóthy, T.: Columbus – Reverse Engineering Tool and Schema for C++. In: Proceedings of the 18th International Conference on Software Maintenance (ICSM 2002). pp. 172–181. IEEE Computer Society (Oct 2002)
- [7] Gyimóthy, T., Ferenc, R., Siket, I.: Empirical Validation of Object-Oriented Metrics on Open Source Software for Fault Prediction. IEEE Transactions on Software Engineering pp. 897–910 (2005)
- [8] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations (2009)
- [9] Heitlager, I., Kuipers, T., Visser, J.: A Practical Model for Measuring Maintainability. Proceedings of the 6th International Conference on Quality of Information and Communications Technology pp. 30–39 (2007)
- [10] International Organization for Standardization: ISO/IEC 15504:2004 Information technology – Process assessment – Part 3: Guidance on performing an assessment. Tech. rep., International Organization for Standardization (2004)
- [11] ISO/IEC: ISO/IEC 9126. Software Engineering – Product quality. ISO/IEC (2001)

- [12] ISO/IEC: ISO/IEC 25000:2005. Software Engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE. ISO/IEC (2005)
- [13] jEdit Homepage. <http://www.jedit.org>, <http://www.jedit.org>
- [14] Jung, H.W., Kim, S.G., Chung, C.S.: Measuring Software Product Quality: A Survey of ISO/IEC 9126. IEEE Software pp. 88–92 (2004)
- [15] Olague, H.M., Etzkorn, L.H., Gholston, S., Quattlebaum, S.: Empirical Validation of Three Software Metrics Suites to Predict Fault-Proneness of Object-Oriented Classes Developed Using Highly Iterative or Agile Software Development Processes. IEEE Transactions on Software Engineering pp. 402–419 (2007)
- [16] Peter Hegedus, Tibor Bakota, Laszlo Illes, Gergely Ladanyi, Rudolf Ferenc, and Tibor Gyimothy: Source Code Metrics and Maintainability: a Case Study. In: In Proceedings of the 2011 International Conference on Advanced Software Engineering And Its Applications (ASEA 2011). pp. 272–284. Springer-Verlag CCIS (Dec 2011)
- [17] The PMD Homepage, <http://pmd.sourceforge.net/>
- [18] Rudolf, F.: Szoftverkarbantartás. Typotex Kiadó (2011)
- [19] SPSS Homepage, <http://www-01.ibm.com/software/analytics/spss/>
- [20] Wagner, S., Lochmann, K., Winter, S., Goeba, A., Klaes, M.: Quality models in practice: A preliminary analysis. In: ESEM '09 Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement. IEEE Computer Society, Washington, DC, USA (2009)

7. Mellékletek

7.1. Alkalmazott metrikák leírása

NOI - Number of Outgoing Invocations

A NOI metódusra adott értéke azt jelenti, hogy hány elemű az a halmaz, amely az összes, a metódusból hívott metódust tartalmazza. Nem tartozik bele az osztályon belüli metódushívás.

LOC - Lines Of Code

Kódsorok száma.

LLOC - Logical Lines Of Code

A metódusban szereplő nem üres, nem komment sorok száma.

NOS - Number Of Statements

A NOS érték a metódus törzsben szereplő utasítások számát adja meg. Nyelvtől függően az alábbi utasítások kerülnek összegzésre: break, continue, do . . . while, empty, for, if, return, switch, while, assert, throw és a kifejezések legmagasabb szintjei.

NLMA - Number of Local Methods Accessed Használt lokális metódusok száma.

NL - Nesting Level

Az NL érték metódusokra határozza meg a vezérlési struktúra maximális mélységét. Csak az if, switch, for, foreach, while és do. . . while szerkezet számít bele. Az NL érték osztályra a metódusai maximális NL értéke.

NFMA - Number of Foreign Methods Accessed

Használt külső metódusok száma.

NII - Number of Incoming Invocations

A NII érték metódus esetén az összes, ezt a metódust meghívó metódusok halmazának a számosságát adja meg.

NUMPAR - Number of Parameters

Metódus esetén a paraméterek számát adja eredményül.

McCC - McCabe Cyclomatic Complexity

A McCabe-féle ciklomatikus komplexitás a metóduson belüli lehetséges futások számát adja meg, amely megegyezik a minimálisan szükséges tesztesetek számával, amennyiben minden lehetséges futást tesztelni szeretnénk. A McCC érték a metóduson belüli döntések száma + 1, ahol minden if, for, while, do . . . while és?: (feltételes kifejezés) elemet egyszer számolunk, minden N lehetőségű switch N+1 értékkel kerül bele, továbbá minden

try block N catch ággal N+1 értékkel kerül be.

CC - Clone Coverage

0 és 1 közötti valós érték, amely kifejezi, hogy a kód mekkora részét fedik le kód másolatok.

PMD

PMD szabálysértések száma [17] a metódusban.

7.2. Vélemények és metrikák közötti korrelációs értékek

9. táblázat. Lineáris kapcsolat erőssége a forráskódmetriák és a szakértői vélemények között

	NOI	LOC	LLOC	NOS	NLMA	NL	NFMA	NII	NPAR	McCC	CC	PMD
Elemezhetőség	-0.44	-0.37	-0.35	-0.33	-0.29	-0.41	-0.37	-0.06	-0.11	-0.28	0.08	-0.27
Változtathatóság	-0.36	-0.37	-0.35	-0.33	-0.18	-0.36	-0.33	-0.04	-0.14	-0.28	0.01	-0.25
Stabilitás	-0.38	-0.37	-0.35	-0.33	-0.24	-0.40	-0.32	-0.09	-0.03	-0.27	0.01	-0.24
Tesztelhetőség	-0.40	-0.35	-0.34	-0.30	-0.20	-0.42	-0.37	0.00	-0.18	-0.28	0.06	-0.25
Érthetőség	-0.39	-0.34	-0.34	-0.32	-0.27	-0.38	-0.32	-0.06	-0.14	-0.27	0.02	-0.26

10. táblázat. Lineáris kapcsolat erőssége a forráskódmetriák és a kiértékelő véleményei között

	NOI	LOC	LLOC	NOS	NLMA	NL	NFMA	NII	NPAR	McCC	CC	PMD
Elemezhetőség	-0.44	-0.62	-0.59	-0.57	-0.28	-0.21	-0.39	-0.05	-0.05	-0.49	0.11	-0.19
Változtathatóság	-0.44	-0.63	-0.60	-0.56	-0.30	-0.19	-0.38	0.00	-0.06	-0.48	0.09	-0.18
Stabilitás	-0.16	-0.46	-0.48	-0.51	-0.20	-0.38	-0.09	0.02	0.02	-0.46	-0.11	-0.31
Tesztelhetőség	-0.44	-0.37	-0.35	-0.33	-0.29	-0.41	-0.37	-0.06	-0.11	-0.28	0.08	-0.27
Érthetőség	-0.37	-0.43	-0.41	-0.38	-0.29	-0.08	-0.30	0.00	-0.10	-0.31	0.09	-0.14

11. táblázat. Lineáris kapcsolat erőssége a forráskódmotrikák és a hallgatói vélemények átlaga között

	NOI	LOC	LLOC	NOS	NLMA	NL	NFMA	NII	NPAR	McCC	CC	PMD
Elemezhetőség	-0.51	-0.54	-0.50	-0.47	-0.35	-0.41	-0.43	0.00	-0.08	-0.37	0.07	-0.38
Változtathatóság	-0.54	-0.50	-0.45	-0.42	-0.36	-0.38	-0.47	-0.03	-0.06	-0.33	0.07	-0.32
Stabilitás	-0.49	-0.46	-0.41	-0.38	-0.28	-0.35	-0.45	0.00	-0.15	-0.29	0.05	-0.30
Tesztelhetőség	-0.47	-0.58	-0.55	-0.51	-0.31	-0.17	-0.41	-0.05	0.00	-0.39	0.06	-0.13
Érthetőség	-0.41	-0.47	-0.43	-0.40	-0.29	-0.39	-0.34	-0.06	-0.06	-0.32	0.05	-0.32
Karbantartathatóság	-0.57	-0.57	-0.52	-0.48	-0.38	-0.43	-0.49	-0.02	-0.14	-0.37	0.10	-0.37

12. táblázat. Logaritmusos kapcsolat erőssége a forráskódmotrikák és a szakértői vélemények között

	NOI	LOC	LLOC	NOS	NLMA	NL	NFMA	NII	NPAR	McCC	CC	PMD
Elemezhetőség	-0.43	-0.44	-0.45	-0.46	-0.28	-0.38	-0.35	-0.04	-0.15	-0.45	0.06	-0.40
Változtathatóság	-0.35	-0.39	-0.40	-0.39	-0.17	-0.35	-0.31	-0.02	-0.18	-0.45	0.034	-0.35
Stabilitás	-0.37	-0.46	-0.46	-0.48	-0.23	-0.38	-0.29	-0.07	-0.07	-0.48	0.02	-0.37
Tesztelhetőség	-0.40	-0.43	-0.44	-0.44	-0.21	-0.41	-0.36	0.01	-0.17	-0.48	0.03	-0.39
Érthetőség	-0.36	-0.36	-0.37	-0.41	-0.26	-0.37	-0.29	-0.05	-0.15	-0.42	0.07	-0.35

13. táblázat. Logaritmusos kapcsolat erőssége a forráskódmotrikák és a kiértékelő véleményei között

	NOI	LOC	LLOC	NOS	NLMA	NL	NFMA	NII	NPAR	McCC	CC	PMD
Elemezhetőség	-0.33	-0.68	-0.68	-0.65	-0.22	-0.40	-0.32	-0.09	-0.11	-0.62	0.34	-0.34
Változtathatóság	-0.33	-0.63	-0.64	-0.59	-0.23	-0.38	-0.31	-0.03	-0.14	-0.59	0.33	-0.34
Stabilitás	-0.08	-0.39	-0.42	-0.42	-0.17	-0.37	-0.05	0.03	-0.01	-0.46	0.09	-0.30
Tesztelhetőség	-0.43	-0.44	-0.45	-0.46	-0.28	-0.38	-0.35	-0.04	-0.15	-0.45	0.06	-0.40
Érthetőség	-0.25	-0.40	-0.41	-0.39	-0.24	-0.22	-0.24	-0.03	-0.18	-0.38	0.23	-0.30

14. táblázat. Logaritmusos kapcsolat erőssége a forráskódmotrikák és a hallgatói vélemények átlaga között

	NOI	LOC	LLOC	NOS	NLMA	NL	NFMA	NII	NPAR	McCC	CC	PMD
Elemezhetőség	-0.49	-0.70	-0.69	-0.69	-0.34	-0.40	-0.42	0.05	-0.12	-0.54	0.13	-0.49
Változtathatóság	-0.53	-0.66	-0.64	-0.64	-0.36	-0.38	-0.46	0.04	-0.10	-0.52	0.17	-0.43
Stabilitás	-0.50	-0.67	-0.65	-0.64	-0.28	-0.36	-0.45	0.03	-0.20	-0.50	0.10	-0.45
Tesztelhetőség	-0.34	-0.62	-0.63	-0.57	-0.26	-0.32	-0.32	-0.09	-0.07	-0.52	0.26	-0.33
Érthetőség	-0.40	-0.62	-0.62	-0.60	-0.28	-0.40	-0.32	-0.03	-0.09	-0.53	0.17	-0.43
Karbantartathatóság	-0.54	-0.75	-0.74	-0.73	-0.37	-0.43	-0.47	0.04	-0.16	-0.58	0.18	-0.51